




Proyecto Final de Módulo

 **Título:** Auditoría, Automatización y Fortalecimiento de la Seguridad en un API RESTful Expuesta a Internet

Objetivo General:

Diseñar, implementar y evaluar un entorno de seguridad automatizado para una API RESTful pública, aplicando estrategias de **mitigación, desarrollo seguro, pruebas de penetración, monitoreo continuo y mejora sostenida** con base en KPI y estándares internacionales.

Escenario Simulado:

La empresa **CyberWise** expone una API RESTful para la gestión de clientes, accesible públicamente. El entorno se encuentra en producción con tráfico activo, pero sin controles de seguridad robustos, ni mecanismos de medición ni monitoreo implementados. El equipo de desarrollo solicita un **diagnóstico exhaustivo**, junto con la implementación de un **plan de mejora técnica y estratégica**.

Actividades del Proyecto:

Parte 1 – Diagnóstico inicial

1. Realiza un análisis de seguridad general del entorno (infraestructura, código, configuración, dependencias).
2. Identifica al menos **cinco vulnerabilidades comunes** (ej: XSS, CSRF, errores 5XX, falta de autenticación robusta, trazas en errores).
3. Justifica técnicamente el riesgo de cada una.

Parte 2 – Automatización del desarrollo seguro (DevSecOps)

4. Diseña un pipeline CI/CD que incluya:
 - Análisis SAST con una herramienta de tu elección
 - Escaneo de dependencias (SCA)
 - Escaneo de contenedor (si aplica)
 - Despliegue controlado con paso de aprobación
 - Reporte automatizado de resultados

Parte 3 – Evaluación continua y métricas

5. Define al menos **cuatro KPI de seguridad relevantes**, incluyendo su fórmula, frecuencia de medición y herramienta de monitoreo.
6. Establece un esquema de auditoría de logs con herramientas como ELK, Wazuh o Splunk.
7. Propón alertas automáticas ante condiciones críticas (ej: más de 20 errores 5XX en una hora).

Parte 4 – Mitigación y reforzamiento

8. Implementa medidas técnicas de mitigación para al menos **tres vulnerabilidades críticas detectadas**.
9. Documenta los cambios aplicados y cómo impactan en la reducción del riesgo.
10. Simula un test de penetración posterior (resumen de resultados reales o estimados).



Criterios de Evaluación (máximo 10 puntos)

1. **Diagnóstico técnico inicial (2 pts)**
Identificación clara y justificada de vulnerabilidades reales del entorno.
2. **Diseño e implementación del pipeline DevSecOps (2 pts)**
El pipeline automatiza tareas clave y demuestra integración de herramientas efectivas.

3. **Definición de métricas e indicadores (1.5 pts)**
KPI bien definidos, medibles y alineados con objetivos de seguridad.
 4. **Implementación de medidas de mitigación (2 pts)**
Soluciones aplicadas de forma efectiva y con evidencia del impacto.
 5. **Simulación de pruebas de seguridad y resultados (1.5 pts)**
Resultados claros y bien documentados tras la mitigación.
 6. **Presentación profesional y claridad del informe (1 pt)**
Redacción clara, estructura lógica y uso adecuado de terminología técnica.
-

Recursos de Apoyo y Herramientas Sugeridas

Seguridad de APIs

- [OWASP API Security Top 10](#)
- [ZAP \(OWASP Zed Attack Proxy\)](#)
- [Postman Security Tests](#)

DevSecOps y CI/CD

- [GitHub Actions](#) / [GitLab CI](#)
- [SonarQube](#) (SAST)
- [OWASP Dependency-Check](#) (SCA)
- [Trivy](#) (Docker scanning)

Monitoreo y Alertas

- [Wazuh](#)
 - [Elastic Stack \(ELK\)](#)
 - [Splunk](#)
 - [Prometheus + Grafana](#)
-



Reflexión Final

“Una API segura no es aquella que nunca falla, sino la que está diseñada para anticipar, detectar y responder éticamente a los fallos. Automatizar la seguridad no solo mejora la eficiencia, sino que profesionaliza la cultura del desarrollo.”
